

## Vordefinierte Parameter

Ein Beispiel für unterschiedliche Umsetzung in einer Programmiersprache bei gleicher Zielrichtung sind die vordefinierten Parameter bei Python.

Es geht dabei um das Ziel, zu einer Klasse Methoden<sup>1</sup> gleicher Aufgabe anzubieten, die man aber mit unterschiedlicher Zahl von Parametern, also nicht nur unterschiedlichen Parameterwerten, aufrufen können will.

### Java

Das Stichwort bei Java ist der **overload**. So möchte man beispielsweise zu einer Klasse einen Standardkonstruktor nutzen, dem beim Aufruf keine Parameter übergeben werden müssen. Gerade für einfache Testzwecke ist das sinnvoll. Andererseits möchte man aber vielleicht auch einen Konstruktor anbieten, mit dem man die zu verwendenden Attributwerte schon beim Aufruf zuweist. Bei Java werden dafür zwei getrennte Konstruktormethoden benötigt.

Ein Beispiel für eine Klasse Rechteck:

```
/**
 * Standardkonstruktor fuer Objekte der Klasse Rechteck
 */
public Rechteck()
{
    x = y = 0;
    w = h = 1;
    farbe = "schwarz";
    gefuehlt = false;
}

/**
 * Konstruktor fuer Objekte der Klasse Rechteck
 * mit uebergabe der initialisierenden Parameter
 * Abmessungen werden nicht uebergeben, werden auf 1 / 1 gesetzt
 */
public Rechteck(double x, double y, String farbe, boolean gefuehlt)
{
    this.x = x;
    this.y = y;
    w = h = 1;
    this.farbe = farbe;
    this.gefuehlt = gefuehlt;
}
}
```

Es ist dadurch ein Aufruf mit `Rechteck rechteck = new Rechteck();` ebenso möglich, wie ein Aufruf `Rechteck rechteck = new Rechteck(10, 50, "red", true);` mit den notwendigen Parameterwerten<sup>2</sup>.

---

1 Das hier für Python Vorgestellte gilt nicht nur für Methoden, sondern auch "normale" Funktionen.

2 Auf die notwendige Eindeutigkeit der Signatur wird hier nicht eingegangen.

### Python

Python hat zu einer Klasse genau einen Konstruktor mit dem Namen `__init__` und löst das Problem durch die Möglichkeit, im Kopf des Konstruktor den angegebenen Parametern einen Standardwert zuzuweisen. Solche Parameter bezeichnet man als Schlüsselwortparameter, englisch *keywordarguments*.

Das entsprechende Beispiel bei Python wäre:

```
class Rechteck():
    def __init__(self, x=0, y=0, farbe='black', gefuehlt=False):
        self.x=x
        self.y=y
        self.farbe=farbe
        self.gefuehlt=gefuehlt
```

Das lässt nun sehr viele Möglichkeiten des Aufrufs zu:

```
r1 = Rechteck()
r2 = Rechteck(10,20)
r3 = Rechteck(farbe='green')
r4 = Rechteck(50, gefuehlt=True)
```

Auch der weniger sinnvolle Fall ist zulässig:

```
r5 = Rechteck(farbe='green', gefuehlt=True, y=20, x=10)
```

Man sollte solche Verwirrungen aber möglichst vermeiden.

### Übung

Schreiben Sie (zumindest) die Köpfe der Methoden (Java und Python) für die Aufgaben, die Rechtecke horizontal zu bewegen, einmal mit zu übergebender Schrittweite und andererseits mit einem Standardwert von 1.

## **Ein Beispiel ohne OO, auch mit optionalen Parametern**

```
# Hilfsfunktion fuer das Beispiel
def quadrat(a):
    return a*a
```

Eine Funktion mit beliebig vielen Parametern

```
def quadrate(a,*args):
    '''Mindestens ein Parameter muss uebergeben werden.'''
    ergebnis=[quadrat(a)]
    for zahl in args:
        ergebnis.append(quadrat(zahl))
    return ergebnis
```

```
print( quadrate(1,2,3,4,5,6) )
```

→ [1, 4, 9, 16, 25, 36]

```
def potenzen(a,*args, **kwargs):
    '''Mindestens ein Parameter muss uebergeben werden.
    vorgegeben ist funktion=quadrat, name="Quadrat" '''
    f,n=quadrat,"Quadrat"
    if len(kwargs)>0:
        f=kwargs['funktion'] # kwargs ist Dictionary
        n=kwargs['name']
    ergebnis=[f(a)]
    for zahl in args:
        ergebnis.append(f(zahl))
    return (n,ergebnis)
```

```
print( potenzen(1,2,3,4,5,6) )
```

→  
( 'Quadrat', [1, 4, 9, 16, 25, 36] )

Mit Übergabe der Funktion und des Namen

```
def kubik(a): return a*3
print( potenzen(1,2,3,4,5,6, funktion=kubik, name='Kuben') )
```

→  
( 'Kuben', [3, 6, 9, 12, 15, 18] )

Einsatz einer anonymen Funktion:

```
print( potenzen(1,2,3,4,5,6, funktion=lambda a:a**5, name='Potenzen 5.Grades') )
```

→  
( 'Potenzen 5.Grades', [1, 32, 243, 1024, 3125, 7776] )